APPLICATION FOR PATENT

TITLE: APPARATUS AND METHOD FOR ADDING LIQUIDITY TO AN ECN AND IMPROVING EXECUTIONS OF ORDERS FOR SECURITIES

INVENTORS: OMAR AMANAT
MICHAEL BUNDY

SPECIFICATION

BACKGROUND

Until just a few years ago, securities were traded only through national and regional exchanges. From the customer's point of view, national exchanges were, and still are, difficult to access, expensive to use, and slow. For a long time, however, national and regional exchanges held an effective monopoly on securities trading. As eventually occurs for all monopolies, competition emerged. Independent electronic communications networks ("ECNs") rose to challenge the exclusive control of the exchanges. ECNs were successful, which encouraged formation of more ECNs. Now ECNs are capable of competing with national exchanges for large-scale trading services.

The lifeblood of ECNs, as for all markets, is liquidity. Market makers traditionally provide liquidity in the form of inventories of securities. ECNs typically are not market makers. The form of liquidity utilized by ECNs is bookings within the ECNs of actual orders for securities. ECNs only exist on the basis of liquidity. More specifically, any individual ECN that wishes to thrive must find ways of providing liquidity in the form of bookings of securities to market.

In prior art, ECNs competed with exchanges in terms of execution quality, especially

speed and transaction cost. Now that there are many ECNs in the marketplace, many of whom provide execution speeds and transaction costs generally superior to that of the large exchanges, often it is true now that ECNs must compete with other ECNs for liquidity. In current art, total round-trip latency between broker-dealer systems and markets ranges from tens of milliseconds to hundreds of seconds, all in a trading environment where markets are often extremely volatile. In these markets, from the customer's point of view, any method of increasing execution speed is highly desirable. Methods and systems for improving order execution quality and methods and systems for generating liquidity for individual ECNs therefore are needed. Moreover, such improved methods and systems benefit the entire marketplace by generally improving both competition for liquidity and improved availability of liquidity.

If an online customer's order flow is directed to a market participant based on latent Level II Quotes, then the customer is at risk of chasing securities. In fact, chasing can occur in any trading situation in which there is substantial delay between changes in actual bookings in a market and the resulting change in a displayed quote. "Chasing" means repeatedly ordering a security at a price that is no longer available because of the delay between the change in the actual quote price or quantity and the display of the quote price or quantity, on the basis of which customers make decisions. An investor who "chases" securities is attempting to buy or sell securities at an order price or in quantities that in fact are no longer available in the market. Some other market participant or investor already bought or sold the securities at the displayed price, and the actual quote price or quantity has changed. The latency in updating quotes results in a display of prices or quantities that are no longer available. Chased orders typically remain unexecuted. It would be useful to have methods and systems for reducing the delay between the time when bookings actually change and the time when new prices or quantities are actually displayed to customers.

SUMMARY

The invention provides methods and systems for improved execution of orders for securities and for adding liquidity to markets. Embodiments include receiving from customers orders for a quantities of securities to be bought or sold, the orders optionally identifying pre-selected markets. Embodiments include sending orders to a first default market where orders are partially filled. Embodiments typically include sending orders to a pre-selected market, where orders again are partially filled, and booking orders in a second default market. Typical embodiments include returning status reports to customers at various stages of order execution.

Typical embodiments include charging fees to customers for execution of orders. Further embodiments include discounting fees charged to customers for orders booked into a second default market. Booking orders into a second default market typically includes setting the order time-in-force to a value other than zero. That is, such bookings typically are non-IOC orders. Orders booked to second default markets therefore will eventually either be partially filled, completely filled, or time out. When such orders are booked, and when such orders eventually fill or time out, embodiments utilizing such bookings will include receiving responses from the second default markets.

In many embodiments, the first default market and the second default market are the same market. In many embodiments, depending on how the default markets are selected, the default markets are sometimes the same market, and sometimes they are different markets.

Further embodiments include selecting, from among a multiplicity of markets, one or more default markets dependent upon default market selection criteria. In many embodiments, default market selection criteria include such factors as transaction costs or access fee levels for execution of orders in markets, response speed of markets (latency),

and liquidity.

In many embodiments at least one of the default markets is connected through tight coupling to a broker-dealer system.

Many markets provide market data feeds to broker-dealers and to customers, typically in the form of quotes. It is the delay between the changes in market information and the provision of the market information to broker-dealers and customers that is one of the causes of chasing. Tight coupling improves the speed of orders and responses, but also improves quote timing for quotes from tightly coupled markets. Tight coupling therefore reduces execution failures from chasing because, for tightly coupled markets, chasing occurs orders of magnitude faster than for non-tightly coupled markets. Market data from a tightly coupled default market, therefore, is much more likely to be current when the market is reached by an order based upon market data in a quote from such a market, and orders based upon such timely quotes are more likely to be filled in that market instead of being chased.

Typical embodiments of the invention send orders first to at least partially execute in a tightly coupled default market before sending the order for further execution to a pre-selected market. This method is used in typical embodiments even for orders whose pre-selected markets were chosen by a customer or a smart execution system. This method is a functional part of the invention because execution against available liquidity in a tightly coupled default market is far superior in terms of speed.

Tight coupling also means that the booking process for orders booked into tightly coupled default markets is far superior in terms of speed. Marketable orders booked because of lack of liquidity in other markets, therefore, are greatly speeded to their display of availability.

In embodiments where close coupling is achieved by installing and operating broker-

dealer and ECN on the same computer system, substantial costs savings result from the need for less computer hardware and reduced system administrative overheads, including, for example, reduced data communications facilities, all of which cost reductions are generally available to benefit the entire market, broker-dealers, ECNs, market makers, and customers. It is a further aspect of many embodiments, especially for the purpose of improving liquidity in a particular default market, to automate discounts, credits, or other forms of payment or credit for orders adding liquidity by booking into a default market. In those instances when broker-dealers through use of the invention can execute or book orders in closely-coupled markets, those orders generally will be executed faster and cheaper than can be done in prior art.

DRAWINGS

Figure 1 is an overview flow diagram of typical embodiments.

Figure 2 is an example data structure for orders.

Figure 3 illustrates selecting default markets by use of market selection criteria.

Figure 4 illustrates aspects of close coupling.

Figure 5 illustrates sending orders to pre-selected markets.

Figure 6 is an overview of embodiments of a system of the invention.

Figure 7 illustrates a system for selecting default markets according to market selection criteria.

Figure 8 illustrates system embodiments of close coupling.

Figure 9 illustrates close coupling through calls to member methods in class objects.

Figure 10 illustrates system embodiments for sending orders to pre-selected markets.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

Definitions:

"Book" or "Booking" refers to an order or sending an order to a market with the order's time-in-force set to a value other than zero, i.e., TIF greater than zero. TIF greater than zero denotes an order that is not an IOC order. Because non-IOC orders, orders with TIF greater than zero, typically are present in a market long enough to be displayed in quotes from the market, the orders are said to be "booked."

"Cancellation" is termination of an order, or partial termination of an order, by the customer or by software comprising an embodiment of the invention. In addition, markets can cancel orders, or parts of orders, for example, in response to an IOC order.

"Customer" refers to any person, trader, or investor, individual, company, or institution, using automated methods and systems for trading, buying or selling, securities.

"Default Market" indicates a market to be used for sending or booking orders regardless of the selection or pre-selection of other markets or when orders partially fill in other markets. In some embodiments, the functions are separated. That is, in some embodiments a first default market is used as a place to send orders regardless whether other markets are selected or pre-selected, and, in the same embodiments, a second default market is used to book orders partially filled in other markets. Orders may partially fill because they become unmarketable or because of a lack of sufficient liquidity in the other markets. In some embodiments, the first default market and the second default market are the same market. In other embodiments, a default market is used only for bookings of orders partially filled first in other markets. Bookings in default markets improve liquidity in those markets. Tightly coupling default markets according to the present invention yields strong improvements in quality of order

execution.

"ECN" abbreviates "Electronic Communications Network," referring to an order matching service that provides liquidity by matching orders rather than by maintaining inventory. In the context of the invention, ECNs are considered markets. ECNs, like market makers are identified by use of market participant identification codes or "MPIDs." In order to avoid confusion with data communications networks, ECNs are referred to as either "ECNs" or as "markets." Some current ECNs, their symbols and names, are listed below. The number and identities of ECNs changes from time to time.

Example List of ECNs	
MPID	**Name**
ARCA	Archipelago
BTRD	Bloomberg Trade Book
INCA	Instinet
ISLD	Island
MWSE	Midwest Stock Exchange
NTRD	NexTrade
REDI	Speer Leeds

"Exchange" means a national, regional, or international exchange for securities trading including for example, Nasdaq or NYSE.

"Executed," in reference to an order, means that shares have been either bought or sold

according to the side of the order.

"Filled" means executed. That is, shares in an order have been executed, bought or sold according to the side of the order. If an order is subject to partial fulfillment, then the order can be partly filled and partly rejected or cancelled. Processing of an order can therefore be completed through some combination of cancellation, rejection, or partial execution. Processing of an order is considered complete when all the shares in the order have been executed, cancelled, or rejected.

"Inside price" means, as appropriate, the highest bid price or the lowest ask price for a particular security. For buy orders, the inside price is the lowest ask price. For sell orders, the inside price is the highest bid price.

"IOC" abbreviates "Immediate or cancel," an order type description meaning that the market to which the order is directed is to fill the order immediately or cancel it.

"Interprocess communications" or "IPC" refers generally to data communications among separate computational processes each of which executes within its own computer address space. In some embodiments, the separate processes execute on separate computers. In other embodiments, the separate processes execute on the same computer. One example of IPC described below is shared memory segments. Other example methods of IPC useful with the invention include pipes, FIFOs (named pipes), message queues, and semaphores. Remote execution methods such as remote procedure calls and uses of CORBA-style object request brokers also are used in some embodiments to pass data among programs or processes operating on the same or separate computers. These methods of IPC, and others as well, are all within the scope of the invention.

"Latency" means a measure of the speed with which markets respond to orders and cancellations. Latency in many embodiments of the invention is determined as the

difference between the time when a response to an order is received and the time when the corresponding order was sent to the market. Latency generally is measured from normal orders, test orders, or test messages. Some markets support test orders or test messages as such. For markets in which test orders or test messages are not supported, tests often are implemented by use of unmarketable orders immediately followed by cancellations. For markets receiving orders regularly, latency typically is tracked from normal orders, without the need for test orders. Latency is embodied as a single ratio difference between two recorded times or as various kinds of averages.

"Level Two Quotes" are quotes that comprise one or more market participant quotes ("MPQs"). The best known source of level two quotes is Nasdaq, but "level two quotes" refers to any form of market information that aggregates market participant quotes for a security.

"Marketable" means limit orders for which the inside price is equal to or better than the order price. That is, Marketable buy orders have order prices equal to or higher than the inside ask price. Marketable sell orders have order prices equal to or lower than the inside bid price. It is helpful to note that the concept of marketability is generally most useful regarding limit orders. That is, market orders as such are inherently marketable, because market orders have no limiting price against which the inside price can be meaningfully compared.

"Market," "electronic market," "market participant," "electronic market participant," "marketing network," and electronic marketing network" are all used as synonyms for services accessible through electronic communications networks capable of executing orders for securities by accepting from broker-dealers buy orders and sell orders, matching or failing to match buy orders with sell orders, and communicating the results to the broker-dealers. Generally the term "market" is used to refer to these entities. All "markets," as the term is used, are either ECNs or market makers. All available markets

have names and symbols as described under the definitions of "ECN" and "market maker."

"Market maker" means a broker-dealer providing order matching and liquidity in a security by maintaining an inventory of the security. Market makers typically trade their inventories through exchanges. Some currently active market makers, their symbols and names, are listed below. The number and identity of market makers can change from time to time.

Example List of Market Makers

MPID	Name
BEST	Bear, Stearns & Co., Inc.
BTAB	Alex, Brown & Sons, Inc.
GSCO	Goldman, Sachs & Co.
HMQT	Hambrecht & Quist, LLC
HRZG	Herzog, Heine, Geduld, Inc.
JANY	Janney Montgomery Scott, Inc.
LEHM	Lehman Brothers, Inc.
MADF	Bernard L. Madoff
MLCO	Merrill Lynch, Pierce, Fenner & Smith Inc.
MOKE	Morgan, Keehan & Co., Inc.
MONT	Nationsbanc Montgomery Securities, LLC

MSCO	Morgan Stanley & Co., Inc.
NITE	Knight Securities, L.P.
OLDE	Olde Discount Corporation
OPCO	CIBC Oppenheimer Corporation
PIPR	Piper Jaffray Inc.
PRUS	Prudential Securities, Inc.
PWJC	Paine Webber, Inc.
RAJA	Raymond James & Associates, Inc.
SBSH	Smith Barney, Inc.
SHRP	Sharpe Capital, Inc.
SHWD	Sherwood Securities Corporation

"MPID" means Market Participant Identifier, a code used to identify all markets, ECNs, and market makers.

"Orders" are orders for purchase or sale of securities. In many of the embodiments described, "orders" are electronic orders for purchase or sale of securities.

"Quotes" are aggregates of information regarding securities traded in markets. Quotes include for securities listed for sale or purchase, symbols identifying the securities, price, side, quantities, and market identifications or MPIDs. Quotes can come from exchanges or directly from markets. A "Nasdaq Level Two Quote" includes market information in the form of market participant quotes for all markets offering to buy or sell a particular security through Nasdaq.

"Securities" are any agreement for investment. Stocks are the securities most often administered in described embodiments of the invention. The invention, however, is applicable to many kinds of securities including, for example, options, commodities, and

bonds.

"Side" refers to which side of the market is represented by an order or a quote. Side indicates whether the quote or order is to buy or sell, bid or ask. "Bid" indicates the buy side. "Ask" indicates the sell side.

"Tight Coupling" means high speed data communications between broker-dealer systems and market systems. In the prior art, total round-trip latency between broker-dealer systems and markets ranged from tens of milliseconds to hundreds of seconds. Tight coupling denotes data communications between broker-dealer systems and market systems at least an order of magnitude faster than was typical in prior art. More specifically, for example, embodiments of the present invention utilizing shared memory segments and semaphores to effect interprocess communications among broker-dealer systems sharing random access memory with one or more market systems will typically demonstrate round-trip latencies of less than one microsecond, a latency improvement in these kinds of embodiments of at least three orders of magnitude over the prior art. Other forms of tight coupling will demonstrate various levels of improvement, but all forms of tight coupling represent improvements in execution quality.

Detailed Description:

Turning to Figure 1, a first aspect of the invention is seen. A first embodiment illustrated in Figure 1 provides a method of executing orders for securities in an automated broker-dealer system (124). This illustrated embodiment includes receiving (102) from a customer (104) an order (106) for a quantity (208) of securities to be bought or sold, the order having an MPID (210) optionally identifying a pre-selected market (116). The illustrated embodiment includes sending (108) the order (106) to a first default market (110), where the order is partially filled (112). The embodiment includes sending (114) the order (106) to a pre-selected market (116), where the order is partially filled (118),

and booking (120) the order (106) in a second default market (122).

Typical embodiments include returning status reports (134) to customers (104) at various stages of order execution (112, 118, 138). Typical embodiments include charging fees (132) to customers (104) for execution of orders. A further embodiment shown in Figure 1 includes discounting (136) fees charged (132) to customers (104) for orders booked (120) into a second default market (122).

Booking (120) orders (106) into a second default market typically includes setting the order TIF (212) to a value other than zero, that is, such bookings typically are non-IOC orders. Orders booked to second default markets therefore will eventually either be partially filled, completely filled, or time out. When such orders are booked, and when such orders eventually fill or time out, embodiments utilizing such bookings will include receiving responses (132) from the second default markets (122).

In a further embodiment, order (106) includes a time-in-force (212), as shown in Figure 1. This kind of embodiment typically includes also setting (130) the time-in-force (212) to "IOC" before sending (114) the order to the at least one pre-selected market (116).

A data structure for orders, useful in many embodiments of the invention is shown in Figure 2. In typical embodiments, as shown in Figure 2, an order (106) includes a symbol (204) identifying securities to be bought or sold, a side (206) indicating whether the securities are to be bought or sold, and a quantity (208) of securities to be bought or sold according to the side. Orders in such embodiments typically include also an MPID (210) optionally set to a market identifier, a time-in-force (212) optionally set to a value greater than zero, and a price (214) optionally set to a value greater than zero.

In other embodiments, the first default market (110) and the second default market (122), shown on Figure 1, are the same market. In many embodiments, depending on how the

default markets are selected, the default markets are sometimes the same market and sometimes different markets.

Turning to Figure 3, a further embodiment is seen to include selecting (306), from among a multiplicity (302) of markets, the default markets (110, 122) dependent upon default market selection criteria (304). In many embodiments, market selection criteria include such factors as (1) transaction costs or access fee levels for execution of orders in markets, (2) response speed of markets, often referred to as 'latency,' the time typically elapsing between sending an order to a market and receiving a response from that market, and (3) liquidity, whether securities generally are available in a market. Various embodiments use alternative default market selection criteria, all of which are well within the scope of the invention.

In still further embodiments at least one of the default markets (110, 122) is connected through tight coupling (126, 128) to the broker-dealer system (124), as shown in Figure 1. Alternative examples of tight coupling useful with the invention are shown in Figure 4. All methods of tight coupling are well within the scope of the present invention.

In typical embodiments, shown in Figure 4, tight coupling includes the capability of interprocess communications of orders (106) and responses (406) to orders through shared memory (408). Forms of shared memory include shared memory segments (410) accessed directly by processes (402, 404) running on a processor, shared files (412) on disk drives or other non-volatile memory, memory configured to emulate a disk drive while operating faster than physical disk drives, or other forms of shared memory. All forms of shared memory are well within the invention. All forms of interprocess communications of orders and responses are well within the invention.

Still further embodiments of tight coupling, as shown on Figure 4, include the capability of communications of orders (106) and responses (406) to orders as parameters in

subroutine calls (414). Similarly, the data communications aspects of the invention in some embodiments are implemented using high speed communications middleware object request brokers developed under the Common Object Request Broker Architecture or "CORBA," the standard for interoperability developed by the nonprofit organization known as the Object Management Group of Framingham, Massachusetts. Some embodiments of tight coupling are implemented by use of remote procedure calls across high speed lines, when communicating across computers, or among processes running on either the same processor or processes running on separate processors on the same computer. CORBA calls also are implemented in some embodiments across high speed lines, when communicating across computers, or among processes running on either the same processor or processes running on separate processors on the same computer.

Also in Figure 4, still further embodiments are illustrated in which tight coupling includes the capability of communications of orders (106) and responses (406) to orders as parameters in calls to class object interface member methods (416). Tight coupling, in embodiments using data communications to send and receive orders and responses between separate broker-dealer systems and market systems, typically includes the capability of communications through directly-connected or networked, dedicated or multipurpose, synchronous or asynchronous, parallel or serial, extremely high speed data communications ports and data communications lines.

In an alternative embodiment shown in Figure 5, sending (114) the order to at least one pre-selected market further includes sending (502) the order to a market (512) identified in the MPID. In embodiments of this kind, the market identified in the MPID often is selected by the customer (104) before the order is received in the broker-dealer system.

In an alternative embodiment also shown in Figure 5, sending (114) the order to at least one pre-selected market further includes sending (502) the order to a market (504) selected by a smart executor (506). A "smart executor" is a system of computer hardware

and software designed and implemented for automated selection of markets for order for securities. An example description of a smart executor useful with the present invention is set forth in Appendix I. The use of any smart executor to select a market for use as a pre-selected market is well within the present invention.

In an alternative embodiment also shown in Figure 5, sending (114) the order to at least one pre-selected market further includes sending (502) the order to a market (514) selected dependent upon a solution set (510) from a solution server (508). "Solution server" refers to automated systems of computer hardware and software capable of generating problem definitions for problems whose solutions depend upon environmental information, receiving the environmental information needed to generate solutions, generating solutions, and communicating the solutions to clients or customers. Solution servers useful with the present invention are any solution servers capable of advance solution of problems of securities trading. An example description of a solution server useful with the present invention is set forth in Appendix II.

Turning now to Figure 6, another aspect of the invention is seen. One embodiment illustrated in Figure 6 provides a broker-dealer system (124) for executing orders for securities. The embodiment of Figure 6 includes a processor (602) programmed to receive (608) from a customer (104) an order (106) for a quantity (208) of securities to be bought or sold, the order having an MPID (210) optionally set to identify a pre-selected market (116). The embodiment further includes a processor (602) programmed to send (614) the order (106) to a first default market (110), where the order is partially filled (616), send (618) the order (106) to at least one pre-selected market (116), where the order (106) is partially filled (620), and book (622) the order in a second default market (122). The embodiment further includes memory (604) coupled (636) to the processor (602), the processor being further programmed to store (638) in the memory (604) the order (106) and programmed to store (640, 642, 646) responses (606) to the order.

Typical embodiments include a processor further programmed to return status reports (612) to customers (104) at various stages of order execution (616, 620, 624). Typical embodiments include the processor programmed to charge fees (610) to customers (104) for execution of orders. A further embodiment shown in Figure 6 includes a processor (602) programmed to discount (630) fees charged (610) to customers (104) for orders booked (622) into a second default market (122). Booking (622) orders (106) into a second default market typically includes the order TIF (212) set to a value other than zero, that is, such bookings typically are non-IOC orders. Orders booked to second default markets therefore will eventually either be partially filled, completely filled, or time out. When such orders are booked, and when such orders eventually fill or time out, embodiments utilizing such bookings will include receiving responses (624) from the second default markets (122).

Also shown in Figure 6 is an embodiment in which the order (106) includes a time-in-force (212). This embodiment further includes the processor (602) programmed to set (628) the time-in-force (212) to "IOC" before sending the order (106) to the at least one pre-selected market (116). In a further embodiment shown in Figure 6, at least one of the default markets (110, 122) is connected through tight coupling (632, 634) to the broker-dealer system (124).

In typical embodiments of the system, as shown in Figure 2, an order (106) includes a symbol (204) identifying securities to be bought or sold, a side (206) indicating whether the securities are to be bought or sold, and a quantity (208) of securities to be bought or sold according to the side. An order includes also an MPID (210) optionally set to a market identifier, a time-in-force (212) optionally set to a value greater than zero, and a price (214) optionally set to a value greater than zero. In various alternative embodiments of the system, the first default market (110) and the second default market (122) are the same market.

A still further embodiment, shown in Figure 7, includes a processor (602) that is programmed to select (702), from among a multiplicity (302) of markets, the default markets (110, 122) dependent upon default market selection criteria (304). In many embodiments, market selection criteria include such factors as (1) transaction costs or access fee levels for execution of orders in markets, (2) response speed of markets, often referred to as 'latency,' the time typically elapsing between sending an order to a market and receiving a response from that market, and (3) liquidity, whether securities generally are available in a market. Various embodiments use various alternative default market selection criteria, all of which are well within the scope of the invention.

In typical embodiments, shown in Figure 8, tight coupling includes the capability of interprocess communications of orders (106) and responses (606) to orders through shared memory. Forms of shared memory include shared memory segments (818) accessed directly by processes running on processors. More specifically, in the illustrated embodiment, a broker-dealer program (402) operating on a broker-dealer processor (602) communicates orders and responses with a market program (802) operating on a market processor (806). More specifically, in the illustrated embodiment, the broker-dealer program (402) is installed in broker-dealer memory (604) coupled (834) to a broker-dealer processor (602) that executes the broker-dealer program, the program (402) installed in memory (604) and operating on the processor (602) comprising a broker-dealer system (800). In the illustrated embodiment, the market program (802) is installed in market system memory (804) coupled (810) to a market processor (806) that executes the market system program, the program (802) installed in memory (802) and operating on the processor (806) comprising a market system (801). The broker-dealer processor (602) and the market processor (806) in the illustrated embodiment are so closely coupled that they share the same memory bus (830), meaning that they are implemented under the same operating system in the same computer. In other embodiments, broker-dealer systems and tightly coupled market systems are installed on separate computers. In still further embodiments, tight coupling is implemented through shared files (820) on disk

drives or through random access memory configured to emulate disk drives while operating faster than physical disk drives. Other forms of shared memory are useful to implement close coupling, and tight coupling through any form of shared memory is well within the invention.

In another embodiment illustrated in Figure 8, tight coupling includes the capability of communications of orders and responses as parameters in subroutine calls (822). In the illustrated embodiment, a broker-dealer system (800) includes broker-dealer system programs (402) installed in broker-dealer system memory (604) and operating on a broker-dealer system processor (602) communicating orders (106) and responses (606) through subroutine calls (822) between the broker-dealer system and a market system (801) that includes market system programs (802) installed in market system memory (804) and operating on a market system processor (806). In some embodiments the market system (802, 804, 806) is installed and operated on the same computer system as the broker-dealer system (402, 602, 604).

Also illustrated in Figure 8 is an embodiment of tight coupling that includes the capability of communications of orders and responses to orders as parameters in calls to class object interface member methods (824). As shown in more detail in Figure 8A, many such object-oriented embodiments include a broker-dealer system (800) programmed to send orders (106) and receive responses (606) from at least one member method (852) in at least one class object (850) in the broker-dealer system (800). Optionally such embodiments include also at least one market system (801) programmed to receive orders (106) and send responses (606) from at least one member method (856) in at least one class object (854) in the broker-dealer system (800).

Tight coupling, in embodiments using data communications to send and receive orders and responses between separate broker-dealer systems and market systems, typically includes the capability of communications through directly-connected or networked,

dedicated or multipurpose, synchronous or asynchronous, parallel or serial, extremely high speed data communications ports and data communications lines. In this context, "separate" means that a broker-dealer system and a tightly coupled market system are not installed and operating on the same computer system to as to utilize memory bus connections to the same physical random access memory. The broker-dealer system and the market system in some embodiments are located in close proximity. In other embodiments, such systems are located remotely from one another.

In a further embodiment shown in Figure 9, a broker-dealer processor (902) is programmed to send (906) orders to at least one pre-selected market (511). In one embodiment of the kind shown in Figure 9, the processor, in sending orders to pre-selected markets (511), is further programmed to send (904) orders to a market (512) identified in an MPID and the market identified in the MPID is selected by the customer (104) before the order is received in the broker-dealer system.

In an alternative embodiment also shown in Figure 9, the processor (902) is programmed to send (906) the order to at least one pre-selected market (511). In this embodiment, the processor is further programmed to send (904) the order to a market (504) selected by a smart executor (506). In the illustrated embodiment, the smart executor (506) is installed and running on at least one separate computer (908). In other embodiments, the smart executor (506) and the broker-dealer system (903) are installed and run on the same computer.

In an alternative embodiment also shown in Figure 9, the processor (902) in a broker-dealer computer system (903) is programmed to send (906) the order to at least one pre-selected market (512, 504, 514). In this embodiment, the processor (902) is further programmed to send (904) the order to a market (514) selected dependent upon a solution set from a solution server (508). In the illustrated embodiment, the solution set is in computer memory (914) on a computer (912) separate from the broker-dealer processor

(902) and separate from the computer (910) on which the solution server (508) is installed. In other embodiments, the solution set (510), the solution server (508) and the broker-dealer system (903) are all installed on the same computer.